Scene classification with Convolutional Neural Networks

Josh King jking9@stanford.edu
Vayu Kishore vayu@stanford.edu
Filippo Ranalli franalli@stanford.edu

1. Introduction

We investigate the problem of *scene classification*, in which scenes from photographs are categorically classified. Unlike object classification, which focuses on classifying prominent objects in the foreground, scene classification uses the layout of objects within the scene, in addition to the ambient context, for classification. The study of scene classification predates the use of convolutional neural networks (CNNs), where, for example, previous techniques include the use of codebooks and bag-of-word models.[3]

In this paper, we examine using supervised learning to train convolutional neural networks to categorize scenes into a predefined set of categories.

2. Problem Statement

We investigate how to train a CNN that can classify scenes with high Top-1 and Top-5 accuracy. We use the Places365-Standard scene dataset, which contains 365 categories with 1.6 million labelled images in the training set, 50 images per category in the validation set, and 900 images per category in the test set [5].

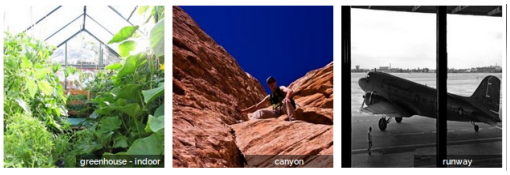

Figure 1. Sample labelled images from the Places2 dataset

However, the test set provided is unlabelled. To evaluate our performance on the test set, we will submit our model's labels to the Places365 evaluation server, which will calculate our Top-1 and Top-5 accuracy. Additionally, the validation data contains only the top label, so we can only evaluate the Top-1 accuracy during cross-validation.

In addition, through our exploration of training CNN's for this purpose, we seek to learn if there are certain neural net features that result in good performance for scene classification.

2.1. Dataset Processing

While we use the entire Places365-Standard dataset, we reduce the resolution of the 256x256 images to 64x64 to allow us to run the images without running out of memory on a Google Cloud Instance.

3. Technical Approach

We train existing CNN's such as ResNet[1] on the dataset. We will then tweak the architecture of the CNN's to see if we can introduce features that will improve performance for scene classification.

In particular, for this report, we have trained the ResNet architecture, which utilizes residual modules which learn residuals between the output of a set of layers and their input. We implement the architecture described in [1] using TensorFlow. Additionally, as suggested in [1], in order to allow a residual layer to span convolutional layers with multiple dimensions, we zero-pad the channel dimension of the output of the layers to maintain a consistent size with the residual. When it comes to residuals between layers of different filter dimensions, we downsample the higher-dimensional residual with a max pool of stride and size of 2. In the future, we plan to address the dimensionality shift in the residual layers with a projection matrix W_s as described in the paper, as this yeilds slightly better performance than padding and downsampling. To verify the correctness of our model, we overfit 100 samples.

4. Preliminary Results

We started by training a 34-layer ResNet written in Tensorflow on the entire downsampled training dataset, validating against the validation set. To train this, we used the hyperparameters outlined in Table 2. Major architectural aspects of our design are described in Table 1. "Residual

Architecture	Description
Loss	Softmax
Optimizer	Adam
Residual Layer Span	2

Table 1. Description of ResNet architecture used for preliminary run

Hyperparameter	Value
Minibatch size	100
Learning Rate	10^{-4}
Gradients Clipped to	5

Table 2. Description of ResNet hyperparameters used for preliminary run

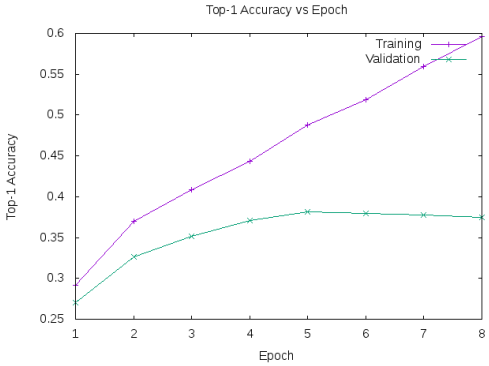

Figure 2. Top-1 Training Accuracy vs Epoch for 34-Layer ResNet

Training	Validation
59.66	37.47

Table 3. Top-1 Accuracy for 34-Layer ResNet after 10 training epochs

Layer Span" describes the number of convolutional layers that the residual layer learns residuals over.

As shown in Figure 2, our preliminary model, with no tuning and initialized to random weights before training obtains over 30% accuracy on the validation set. We expect further refinement of this model to allow us to boost this further. The leaderboard of the Places365-Standard dataset currently reports a 50% best Top-1 accuracy. We also trained a smaller, 18-layer ResNet (shown in Figure 3 and Table 4), for which we obtained similar results, with a training accuracy of around 60%, and a validation accuracy of 36% .

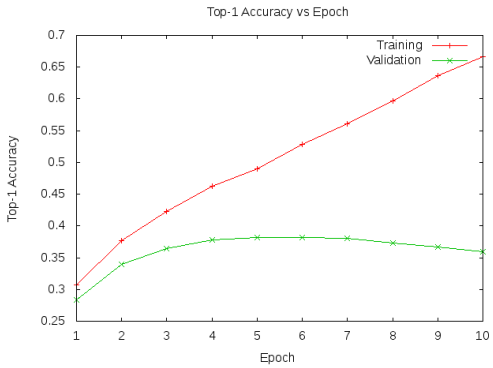

Figure 3. Top-1 Training Accuracy vs Epoch for 18-Layer ResNet

Training	Validation
66.4	35.90

Table 4. Top-1 Accuracy for 18-Layer ResNet after 10 training epochs

5. Discussion

Based on our results, though we perform well on the test set, our performance on the validation set is much lower due to overfitting. To address this, we will try adding regularization, such as through dropout layers, and tuning the learning rate further. In addition, we also noticed that our validation accuracy decreased during our later training epochs while the training accuracy continued to rise. To address this, we will add an early stop to terminate training. Furthermore, we expect to obtain stronger results after tuning other hyperparameters within the CNN's.

6. Future Work

Our current setup provides us a foundation on which we will test further modifications and analyze how they improve or detract from our accuracy on the scene classification problem. Futhermore, we plan on visually examining misclassified images, to gain more insight into reasons why our CNN's may be unable to categorize them properly. Future avenues of exploration may also include applying transfer learning by using CNN's with pre-trained weights for object classification, and retraining the final set of fully connected layers on the scene classification dataset. We also plan to investigate further forms of regularization such as dimensionality reduction through autoencorders. Finally, if time permits, we will explore an ablation study of our models to evaluate if there are certain characteristics of a CNN that allows it to perform well on scene classification.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[2] D. Lin, C. Lu, R. Liao, and J. Jia. Learning important spatial pooling regions for scene classification. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3726–3733, June 2014.

[3] N. Rasiwasia and N. Vasconcelos. Scene classification with low-dimensional semantic spaces and weak supervision. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[4] R. Wu, B. Wang, W. Wang, and Y. Yu. Harvesting discriminative meta objects with deep cnn features for scene classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1287–1295, Dec 2015.

[5] B. Zhou, A. Khosla, À. Lapedriza, A. Torralba, and A. Oliva. Places: An image database for deep scene understanding. *CoRR*, abs/1610.02055, 2016.